

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ONLINE EDITOR OBRÁZKŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAKUB VODÁK

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ONLINE EDITOR OBRÁZKŮ

ONLINE IMAGE EDITOR

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAKUB VODÁK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. PAVEL SVOBODA

BRNO 2011

Abstrakt

Cílem této práce je vytvořit aplikaci pro online editaci statického obrazu. Jde především o přívětivé uživatelské rozhraní a jednoduchost. K dispozici jsou základní grafické operace. Aplikace umožňuje vytváření uživatelských účtů a s tím spojenou možnost manipulace a sdílení obrázků. Je využito prostředí Adobe Flash a programovacího jazyka ActionScript 3.0.

Abstract

The purpose of this thesis is create application for online edition of static image. The main objective is user-friendly interface and simplicity. Basic graphic operations are available. The application allows you to create user accounts and the related ability to manipulate and share images. It is based on Adobe Flash and programming language ActionScript 3.0.

Klíčová slova

Editace, Úpravy, Obraz, Fotky, Filtry, Transformace, Kreslení, Adobe Flash, ActionScript 3.0, Sdílení, Web

Keywords

Editing, Modify, Image, Photos, Filters, Transformation, Painting, Adobe Flash, ActionScript 3.0, Sharing, Web

Citace

Jakub Vodák: Online editor obrázků, bakalářská práce, Brno, FIT VUT v Brně, 2011

Online editor obrázků

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Pavla Svobody.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jakub Vodák
18. 5. 2011

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce, panu Ing. Pavlu Svobodovi za spolupráci a vedení. Dále Ing. Karlu Sekotovi DiS. za rady a motivaci po celou dobu studia. Dále také své rodině, která mě vždy podporovala.

© Jakub Vodák, 2011

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
2 Existující řešení	4
2.1 Picnik	4
2.2 Pixlr.....	4
3 Používané technologie	5
3.1 Webová aplikace.....	5
3.2 Klientská část.....	5
3.3 Serverová část.....	6
4 Multimediální obsah	8
4.1 Adobe Flash.....	8
4.2 Microsoft Silverlight	8
4.3 HTML 5	8
4.4 Srovnání	9
5 Zpracování obrazu	10
5.1 Transformace.....	10
5.2 Filtrace	10
6 Návrh	13
6.1 Uživatelské rozhraní	13
6.2 Model případů užití.....	15
6.3 Entity Relationship Diagram	16
7 Implementace	17
7.1 Diagram tříd.....	17
7.2 Výchozí zobrazení	18
7.3 Komunikace se serverem.....	19
7.4 Problémy s řešením.....	19
8 Výsledek.....	20
8.1 Zhodnocení uživatelského rozhraní	20
8.2 Testování	21
8.3 Porovnání	25
9 Závěr	26
Literatura	27
Seznam příloh	28
Příloha 1	29

Příloha 2 30

1 Úvod

Editace obrazu je důležitou částí počítačové grafiky, se kterou se setkáváme na každém kroku. Od klasické úpravy fotografií v našem rodinném albu, přes stránky časopisů, až po velké reklamní plochy. Je zkrátka tendencí statický obraz upravit – vylepšit, případně něco doplnit nebo odstranit. Existuje řada grafických počítačových programů, které nabízejí mnoho možností. Ovšem trendem současné doby je přenášení osobních dat a aplikací do světa on-line tak, aby byly přístupné kdykoliv a kdekoliv bez závislosti na konkrétním počítači.

Proto je téma Online editor obrázků aktuální. Samozřejmě takové editory již existují a jsou na poměrně vysoké úrovni. Na internetu se ovšem pohybuje široké spektrum uživatelů a vyskytuje mnoho druhů aplikací, které nějakým způsobem využívají fotografie, respektive grafický obraz. Při zaměření se na určitou skupinu tak lze vytvořit aplikaci, která bude originální a dobře použitelná.

Cílem této práce je tedy popsat průběh vývoje takového editoru a všech jeho souvislostí. Po výběru tohoto zadání bylo prvním cílem zvolit vhodné technologie a naučit se je používat. Dále bylo třeba seznámit se s teoretickým základem pro zpracování obrazu, i když součástí práce není jeho přímá editace, ale využívání různých knihoven a modulů. Návrh aplikace se kromě samotného editoru zabývá širším uživatelským rozhraním pro správu obrázků. V následné implementační části jsou uvedeny specifické prvky aplikace a také komplikace, které vznikaly při řešení. Dále následuje zhodnocení výsledné aplikace, její testování skupinou osob, porovnání s jinými editory a v závěru možné budoucí rozšíření.

Aplikaci jsem vytvářel na vlastním webovém serveru a je dostupná na adrese <http://bp.jakubvodak.cz>, kde se nachází úvodní stránka. Ta plní roli prezentace celého projektu a je vstupním bodem do editoru. Obsahuje zásuvný modul sociální sítě Facebook pro sdílení.

2 Existující řešení

Při hledání již fungujících editorů není potřeba velkého úsilí. Po zadání fráze „*online photo editor*” do některého vyhledávače, lze ihned najít několik editorů. V podstatě by se daly rozdělit do dvou skupin. První, které se podobají programu Photoshop a druhé, které se snaží zaujmout originálním řešením. Podstatné rozdíly jsou také v kvalitě zpracování. Mezi ty nejpoblárnější patří Picnik¹ a Pixlr².

2.1 Picnik

Editor s netradičním uživatelským rozhraním, aktuálně ve vlastnictví společnosti Google. Na první pohled jde zcela jinou cestou než je běžné u většiny podobných aplikací – žádné klasické menu a panel nástrojů. Je navržen tak, aby ho mohl plně využívat i méně zkušený uživatel. Operace jako je změna jasu a jí podobné funkce, lze nechat vyladit automaticky. Obsahuje mnoho filtrů a kreativních nástrojů – např. vytváření pohlednic podle předloh, blahopřání, rámy a další.

Umožňuje vytváření uživatelských účtů. Pro funkce jako jsou křivky, úrovně nebo klonovací razítko je potřeba vytvořit placený účet. Registrace vyžaduje základní povinné údaje, zbytek údajů je dobrovolných. Možnosti ukládání a sdílení jsou rozšířené na všechny známé sociální sítě a webová alba. Další možností je obrázek přímo odeslat na libovolný email.

2.2 Pixlr

Tento editor lze nazvat online alternativou pro známý program Photoshop. Obsahuje plnohodnotné kreslení, práci s vrstvami, mnoho filtrů a nástrojů. Je zpracován velmi profesionálně, disponuje rychlou odezvou a nabízí jazykovou lokalizaci do téměř 30 jazyků včetně češtiny. Práci s ním ocení především pokročilejší uživatelé, kterým umožní editaci na nejvyšší možné úrovni.

Rovněž umožňuje vytváření uživatelských účtů. Registrace je velice jednoduchá a vyžaduje pouze základní povinné údaje jako přihlašovací jméno, heslo a email. Uživatel tímto získá vlastní úložný prostor pro své obrázky. K dispozici je propojení se sociální sítí Facebook a webovými alby Picasa a Flickr.

¹ viz <http://www.picnik.com/>

² viz <http://pixlr.com/>

3 Používané technologie

Při tvorbě internetových stránek se dnes využívá celá řada technologií a programovacích jazyků. Kvalitní webová prezentace totiž musí umět zaujmout na první pohled a přitom nabízet všechny potřebné informace přijatelným způsobem. Na druhé straně musí být dobře technicky realizovaná, lehce modifikovatelná a zabezpečená. Pokud se některé části stránky zobrazují nekorektně nebo z důvodu špatné funkčnosti jsou zcela nepřístupné, vyvolávají v uživateli dojem nedůvěryhodnosti a neprofesionality. Zajištění výše zmíněných požadavků lze dosáhnout právě vhodným kombinováním a spoluprací různých prostředků, které zde budou popsány. *(volně převzato z [2])*

U jednotlivých technologií bude také uvedeno jejich konkrétní využití v tomto projektu. I když to nebylo přímo součástí zadání, snažil jsem se použít co nejvíce prostředků a poznat mezi nimi pro mne dosud neznámé souvislosti.

3.1 Webová aplikace

„...je aplikace poskytovaná uživateli z webového serveru přes počítačovou síť internet.“

Hlavní výhoda tedy spočívá v její dostupnosti, možnosti přistupovat k ní bez potřeby cokoliv instalovat, nezávislosti na architektuře nebo operačním systému. Nemá žádnou údržbu, či aktualizace a v základních verzích jsou většinou k dispozici zdarma. Nevýhodou je závislost na internetovém připojení a poskytovateli této aplikace, který k ní může omezit nebo zcela zrušit přístup. V přímém porovnání s lokálně nainstalovaným programem se tváří méně robustně, což je celkem logický důsledek.

Pokud aplikaci tvoří jeden celistvý program, je její struktura popisována jako vícevrstvá architektura (nejčastěji 3 vrstvy). První prezentační vrstvou je prohlížeč, který pracuje na straně klienta, zobrazuje uživatelské rozhraní a stará se o zadávané vstupy. Střední logická vrstva pracuje na serveru, zpracovává data a může např. dynamicky generovat obsah stránek. Třetí vrstvu tvoří nejčastěji databáze, kde se uchovávají a zpřístupňují data. [1]

3.2 Klientská část

Je zastoupena webovým prohlížečem, který zobrazuje uživatelské rozhraní a rovněž GUI (*Graphical User Interface*) stránky. Častým problémem je nejednotnost prohlížečů při jejich zobrazování a pak i v některých skutečnost, že uživatel může nastavením vypnout, či zablokovat některé potřebné moduly a funkce. *(Tato podkapitola vychází z [2])*

3.2.1 XHTML

XHTML (*eXtensible Hypertext Markup Language*) je značkovací jazyk pro tvorbu hypertextových dokumentů vycházejících z jazyka HTML. První specifikace jazyka XHTML si dávala za cíl převod staršího jazyka HTML do nové podoby tak, aby vyhovovala podmínkám tvorby XML dokumentů při zachování zpětné kompatibility. Každý XHTML dokument je tedy validním XML dokumentem. Je dostupný ve třech variantách – Strict, Transitional a Frameset. V projektu jsem použil XHTML verze

1.0 ve variantě Transitional, aby bylo možné použít prostředky z předchozích specifikací v souvislosti s dalším kontextem aplikace. Stránka je ověřena jako validní na W3C³.

3.2.2 CSS

CSS (*Cascading Style Sheets*) neboli kaskádové styly jsou deklarativním jazykem pro definici jednotného vzhledu stránek. Důvodem jeho zavedení byla potřeba oddělit formu a samotný obsah. Výhodou takového přístupu je možnost měnit vzhled bez zásahu do obsahu, zachování konzistence a větší možnosti formátování. Použití v projektu je však zcela minimální, pouze pro zarovnání stránky a nastavení barvy pozadí.

3.2.3 JavaScript

Jedná se o interpretovaný, objektově orientovaný skriptovací jazyk (syntakticky podobný C/C++), určený k tvorbě programů běžících ve webovém prohlížeči. Ačkoliv se jedná o poměrně plnohodnotný programovací jazyk, je využíván spíše pro doplňkové prvky jako např. kontrola formulářových polí. Pro svou možnost interaktivity se jevil jako potenciální pro tento projekt, ale nakonec bylo využito jiných technologií (viz další bod).

3.2.4 RIA

RIA (Rich Internet application) v překladu bohatá internetová aplikace je typická pro své vlastnosti podobné těm desktopovým. Do stránky se vkládá externě a o její běh se stará prohlížeč. Typickými zástupci jsou Adobe Flash, nebo Microsoft Silverlight a právě první jmenovaný je hlavním stavebním kamenem tohoto projektu (viz kapitola 4).

3.3 Serverová část

Webový server umožňuje přístup k obsahu stránek a dalších dat pomocí protokolu HTTP. Reakcí na požadavky klienta může být statický obsah, kdy dojde pouze k předání požadovaných dat. Druhou možností je dynamický obsah, kdy na základě požadavků jsou data nejdříve zpracována a předána v konkrétní podobě. (*Tato podkapitola vychází z [6]*)

3.3.1 PHP

PHP (Hypertext Preprocessor) je skriptovací programovací jazyk (syntaxí vychází z Perlu a C), určený především pro tvorbu dynamických internetových stránek. Skripty probíhají na straně serveru a klientovi se předává pouze interpretovaný výstup. Podporuje databázové systémy a poskytuje mnoho předdefinovaných funkcí např. pro zpracování textu. Jedná se o jeden z nejrozšířenějších jazyků pro web a je všeobecně uznávaným standardem. Použití v tomto projektu (verze 5.2.17) je poměrně rozsáhlé a je mu věnována kapitola 7.3.

³ ověřeno na <http://validator.w3.org>

3.3.2 MySQL

Jedná se o multiplatformní relační databázový systém, založený na transakčním dotazovacím jazyku SQL. Proti ukládání dat přímo do souborů na serveru má použití databáze několik výhod jako efektivnější práci s velkým množstvím strukturovaných dat nebo omezení bezpečnostních rizik při povolení zápisu na lokální souborový systém. MySQL server je vyvíjen jako software s otevřeným kódem, vyniká pro svou rychlost a jednoduchost a nejčastěji využívaný je právě v kombinaci s jazykem PHP. Návrh databáze (verze 5.0.90) je popsán v kapitole [6.1.2](#).

4 Multimediální obsah

Požadavky na obsah internetových stránek neustále rostou a v dnešní době je zcela běžné přehrávání videa, interaktivní grafiky, animace nebo hry. K tomu již nestačí základní technologie jako je např. JavaScript. Většinou se pro zobrazení takového obsahu využívá různých plug-inů pro prohlížeč. Jistou revoluci postupně přináší jazyk HTML 5.

4.1 Adobe Flash

Nejúspěšnější plug-in pro prohlížeče. Na jeho počátku stál kreslicí program SmartSketch, který byl v roce 1995 obohacen o snímkovou animaci a přejmenován na FutureSplash Animator. O rok později je odkoupen firmou Macromedia a sloučen s podobným projektem Shockwave. Stále se však jednalo pouze o kreslicí a animační prostředí. Prvky interaktivity přidal až jazyk ActionScript inspirovaný jazyky Java a C++. V roce 2005 kupuje Macromedii firma Adobe Systems a pokračuje s vývojem ve verzích označených CS.

Jedná se tedy o multimediální platformu často souhrnně označovanou jako RIA (viz 3.2). Umožňuje vytvářet interaktivní obsah, který běží na straně klienta a podporuje dynamické grafiky, média a animace. V praxi se nejčastěji využívá k tvorbě bannerové reklamy, her a přehrávačů videa. Formát výsledného souboru SWF (ShockWave Flash) potřebuje ke spuštění interpret, většinou v podobě již zmíněného plug-inu. [2]

4.2 Microsoft Silverlight

Přímý konkurent firmy Adobe představený firmou Microsoft v roce 2006. Svou podstatou velmi podobný produktu Flash, nicméně se několika prvky odlišuje. Hlavním rozdílem je skutečnost, že je založený na zredukované verzi .NET⁴. Umožňuje tak vývojářům psát kód na straně klienta v C# a využívat stejných abstrakcí jako na straně serveru. Zaměřuje se také na integraci do mobilních zařízení Windows Phone 7.[7] Blíží se vydání páté verze, která bude obsahovat například podporu 64-bitových systémů nebo lepší využití hardwarové akcelerace.

4.3 HTML 5

Internetové stránky využívají pro definici svého obsahu jazyk HTML verze 4 nebo XHTML 1.1. Kromě toho jsou však potřeba i další jazyky a technologie (viz 3.2). Tento stav má právě HTML 5 změnit a stát se mocnějším nástrojem v rukou vývojářů. Jeho vývoj začal v roce 2004 a ještě několik let bude trvat (první finální verze má být k dispozici příští rok). Již nyní ale známe některé prvky a můžeme najít ukázkové aplikace. [4]

- *Strukturování stránky* – byla a je záležitostí kaskádových stylů (CSS). Nová specifikace HTML však přináší nové elegantnější řešení v podobě značek, které jsou přímo zaměřené na vytváření struktury stránky.

⁴ více na <http://cs.wikipedia.org/wiki/.NET>

- *Multimediální obsah* – již nebude potřeba využívat prvky třetích stran např. pro přehrávání videa. Přibudou značky *video* a *audio* a o samotné přehrávání se postará prohlížeč.
- *Vykreslování grafiky* – asi největší novinkou bude značka *canvas*, schopná vykreslovat grafiku. To umožní např. dynamicky vykreslovat grafy z aktuálních dat a pozadu nezůstane ani v oblasti 3D grafiky. Další možností je využití programů (např. Wallaby) pro převod aplikace vytvořené v prostředí Flash do HTML 5.
- *Více vláken* – tzv. Web Workers je technologie postavená na JavaScriptu a umožňuje vykonávat několik operací zdánlivě zároveň.
- *a další...*

Nová specifikace jazyka HTML nebyla součástí zadání a je zde uvedena pouze informativně. Proto nebude dále porovnávána ani jinak využita.

4.4 Srovnání

Účelem tohoto srovnání je výběr vhodné technologie pro tento projekt. Platformy Adobe flash i Microsoft Silverlight jsou neustále ve vývoji, a proto jsem se snažil najít co nejaktuálnější informace především na oficiálních stránkách. Závěrem srovnání je fakt, že nelze jednoznačně určit, která technologie je výhodnější. Z tohoto důvodu jsem zvolil produkt firmy Adobe na základě svých dřívějších zkušeností.

	<i>Adobe Flash</i>	<i>Microsoft Silverlight</i>
Aktuální verze	10.0.22.87	4.0.60310.0
Velikost plug-inu	2,7 MB	6 MB
Rozšíření (odhad)	95%	70%
Programovací jazyky	ActionScript	Common Language Runtime
Zpracování vyhledávačem (SEO)	pouze Google	všechny
Ochrana obsahu (DRM)	Flash Access	PlayReady

Tabulka 2.1: Porovnání některých vlastností platforem Flash a Silverlight

Počet rotujících objektů v animaci	<i>Adobe Flash</i>		<i>Microsoft Silverlight</i>	
	FPS	Využití CPU	FPS	Využití CPU
3	60	10%	100	15%
30	60	15%	100	15%
60	50	55%	57	75%
100	30	65%	25	75%
Testovací PC	Core 2 Duo 2.53GHz, 4GB RAM, Windows 7, Google Chrome			

Tabulka 2.2: Porovnání výkonu platforem Flash a Silverlight při jednoduché animaci⁵

⁵ animaci lze vyzkoušet zde <http://www.shinedraw.com/mathematics/flash-vs-silverlight-fps-meter-stress-test/>

5 Zpracování obrazu

Jedná se o obor, který se zabývá analýzou obsahu obrazu. Takový obraz můžeme chápat jako spojitou funkci dvou proměnných nebo 2D spojitý signál. Nejčastěji bývá reprezentován jako „rastr“ (matice obrazových bodů) získaný vzorkováním. Jednotlivé body (pixely) se rovněž vzorkují, a to na rozsah úrovní šedé 0-255, 0-65535, případně jako RGB model 3x 0-255. (Tato kapitola vychází z [5])

5.1 Transformace

Geometrické transformace se používají pro zvětšování, posouvání nebo třeba otáčení obrazu. Každý jeho bod o souřadnicích (x_1, y_1) mapují na výstupní pozici (x_2, y_2) . Pro tyto transformace se používají homogenní souřadnice, jejichž myšlenkou je reprezentovat bod ve vektorovém prostoru o jednu dimenzi větším. Bod $[x, y]^T$ se vyjádří ve 3D vektorovém prostoru jako $[\lambda_x, \lambda_y, \lambda_z]^T$, kde $\lambda \neq 0$. Pro jednoduchost se však používá některé z vyjádření $[x, y, 1]^T$. Důvodem je sjednocení výše zmíněných operací.

Při aplikaci více transformací záleží na jejich pořadí, protože v konečné podobě lze vše reprezentovat jedinou maticí. Té dosáhneme postupným násobením jednotlivých matic jediné ve správném pořadí. Dále platí, že pokud násobíme transformační matici, maticí k ní inverzní, dostaneme původní obraz.

Pro získání hodnot obrazové funkce mimo naměřené diskrétní hodnoty (např. při zvětšení obrazu) se používají interpolační metody. Mezi ty běžně používané patří:

- *Metoda nejbližšího souseda* – přiřadí bodu (x, y) hodnotu jasu nejbližšího sousedního bodu. Výhodou je rychlost a stálost barev.
- *Bilineární interpolace* – používá čtyři okolní body, přičemž s rostoucí vzdáleností každého bodu klesá jeho vliv. Jedná se o použití lineární metody ve dvou osách.
- *Bikubická interpolace* – vezme z okolí 16 bodů a proloží je bikubickým polynomem. Výhodou je vyhlazenější obraz obsahující méně interpolačních artefaktů. To vše za cenu nižší rychlosti.

5.2 Filtrace

Přesněji řečeno nás bude zajímat lineární filtrace ve smyslu platnosti principu superpozice (reakce na součet dvou a více signálů) nad vstupy a výstupy filtrů. Tím lze dosáhnout změny nebo zvýraznění některých složek signálu. Základní operací při filtraci obrazu je diskrétní konvoluce, kterou můžeme popsat následujícím vztahem, kde funkce $f(x)$ zastupuje původní obrázek a funkce $g(x)$ se nazývá konvolučním jádrem.

$$(f \times g)(m) = \sum_n f(n)g(m - n) \quad (5.1)$$

Filtr postupně zpracovává všechny body obrazu a pro každý z nich vynásobí hodnotu aktuálního bodu a osmi jeho sousedních bodů odpovídajícími hodnotami konvolučního jádra. Výsledné hodnoty poté sečte a přiřadí aktuálnímu bodu. Výpočet takových jader je předmětem matematických výpočtů, hotová jádra jsou však známá. (*Příklady filtrů vycházejí z [8], obrázky vytvořené editorem tohoto projektu byly doplněny na závěr.*)

5.2.1 Zaostrění

Pokud máme rozostřený obraz nebo všeobecně obraz bez ostrých hran, lze využít filtr pro ostření, který hrany zvýrazní. Nevýhodou je zvýraznění nechtěných prvků, jako je šum. Příklady konvolučních jader:

Původní rozostřený obrázek

$$H1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

$$H2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$



Obrázek 5.1 Rozostřený



Obrázek 5.2 Zostření 1



Obrázek 5.3 Zostření 2

5.2.2 Detekce hran

Hranou se v digitálním obraze rozumí místo, kde se prudce mění jas. Tato operace souvisí s doostřováním a zlepšuje vnímání lidského zraku. V praxi nachází široké uplatnění v rozeznávání objektů, jako jsou např. otisky prstů.

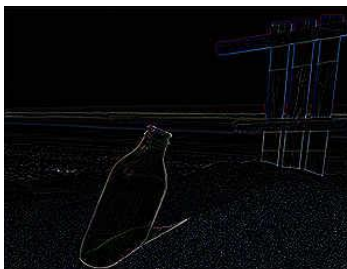
Původní obrázek

$$H3 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$H4 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



Obrázek 5.4 Původní



Obrázek 5.5 Detekce hran 1



Obrázek 5.6 Detekce hran 2

5.2.3 Rozostření

Zde se využívá vyhlazovacích filtrů, které zprůměrují okolní body, čímž se dosáhne rozmazání ostrých barevných přechodů. To značně ovlivní tenké čáry a některé detaily.

Původní obrázek

$$H5 = \frac{1}{9} \times \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$H6 = \frac{1}{16} \times \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$



Obrázek 5.7 Původní



Obrázek 5.8 Rozostření 1



Obrázek 5.9 Rozostření 2

5.2.4 Reliéf

Forma zkreslení, kdy plocha obrazu vystupuje do prostoru. Světlé oblasti jsou vyzdviženy, zatímco tmavé naopak zatlačeny do obrazu. Směr osvětlení lze nastavit pootočením matice.

Původní obrázek

$$H7 = \begin{pmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

$$H8 = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -2 \end{pmatrix}$$



Obrázek 5.10 Původní



Obrázek 5.11 Reliéf 1



Obrázek 5.12 Reliéf 2

6 Návrh

Tato kapitola popisuje způsob návrhu aplikace a její součástí. Představím zde také základní trendy, které se využívají při vytváření uživatelských rozhraní. Zadání projektu požaduje návrh s důrazem na efektivnost a uživatelskou přívětivost. Po prohlédnutí již existujících řešení (viz 2) je takřka nemožné vyrovnat se jim po technické stránce, proto mám v plánu přidat alespoň originální způsob využití.

6.1 Uživatelské rozhraní

Ve spojení s výpočetní technikou se často sleduje produktivita neboli schopnost člověka tuto techniku efektivně používat. Nacházíme se v době, kdy výrazně vzrostl výkon počítačů a výrazně poklesly jejich rozměry. Navíc díky nízké ceně jsou dostupné téměř pro každého. Z toho vyplývá snaha přizpůsobit počítače člověku tak, aby s nimi mohl snadno komunikovat. Přitom je potřeba brát zřetel na různé schopnosti uživatele. I méně zdatný člověk by měl být schopen aplikaci používat, zatímco pokročilejšímu umožní efektivní práci. *(volně převzato z [3])*

Aplikace by měla být rozdělená na úvodní stránku prezentující projekt a samotný editor, jak je tomu u všech podobných aplikací. Protože nejsem příliš zdatný grafik a vytvořit z bílé stránky zajímavý návrh není úkol pro každého, začal jsem hledáním nějaké předlohy, která by dala celé práci myšlenku. Zaujal mne obrázek dvou složek a několika volně položených fotografií. Do složek by se vkládaly nové fotky a následně i ukládaly ty z editoru. To nabízí mnoho možností na jejich sdílení. Vše bych potom rozdělil na veřejnou a soukromou část (viz Obrázek 6.1). Součástí návrhu tedy není pouze samotný editor, ale i rozhraní pro správu fotek. To zahrnuje vytváření uživatelských účtů, ukládání informací do databáze a způsob manipulace s fotkami na serveru.



Obrázek 6.1: Návrh organizace fotek

6.1.1 Vektorová grafika

Jak již bylo uvedeno dříve, projekt je vytvářen v prostředí Flash, kde primární formou zobrazování je vektorová grafika. Celé prostředí jsem proto ručně překreslil do vektorové podoby. Jistou pomocí je vestavěná funkce *Trace Bitmap*, která podle zadaných parametrů konvertuje bitmapový obrázek do vektorové podoby. Nicméně v případě velice členitého obrázku není výsledek nijak oslnivý a slouží spíše jako šablona.

V případě úvodní stránky se po úspěšném převodu u výsledného *swf* souboru zmenšila velikost o 300% a ve finální verzi zabírá pouze 100KB. Další velkou výhodou je přizpůsobení se velikosti okna prohlížeče, potažmo velikosti monitoru.

6.1.2 Uživatelské účty

Každému návštěvníkovi je umožněna registrace, při níž jsou požadovány pouze základní údaje jako je jméno, heslo a email. Zatímco nepřihlášený uživatel může použít výše zmíněné složky pouze veřejně, přihlášený může zvolit mezi veřejnou a soukromou formou. V případě veřejné formy je obsah složek volně přístupný komukoliv. Z popsaného konceptu vzniká originální myšlenka, a to možnost, nechat svůj obrázek upravit někým jiným. Pro názornost několik referenčních příkladů, které odpovídají následnému modelu případů užití (Obrázek 6.2):

- *Člověk A* nahraje veřejně fotku do složky **neupravených** fotek (do složky upravených fotek lze vkládat pouze výstupy z editoru). *Člověk B* si fotku může prohlédnout, upravit a následně uložit do složky **upravených** fotek (nelze uložit zpět do neupravených). *Člověku A* přijde upravená fotka na email (viz. funkce Upshare) a je dále dostupná ve složce upravených fotek.
- *Člověk A* nahraje soukromě fotku do složky neupravených fotek. Tato fotka je dále pro editaci přístupná pouze jemu.
- *Člověk A* otevře veřejnou/soukromou fotku v editoru ze složky upravených/neupravených fotek a uloží ji do svého počítače.
- *Člověk A* nahraje fotku přímo do editoru a po editaci uloží zpět do svého počítače (nulové využití složek).

Funkce Upshare

Jedná se o novou funkci, která rozšiřuje možnosti sdílení. Každý obrázek, který uživatel nahraje pod svým účtem do veřejné složky (pro všechny), bude mít stále pod kontrolou. Pokud si tento obrázek kdokoliv otevře a jakkoliv upraví, přijde dotyčnému následně na email.

6.1.3 Správa Fotek

Obrázky uložené na serveru jsou rozděleny do několika adresářů. Při vyládání přímo do editoru je využit adresář *temp*, jak už název napovídá, jedná se o dočasné úložiště. Obrázky zde zůstávají i po jejich zavření v editoru a jsou posléze smazány automatickým skriptem. Dále adresář *public* slouží pro manipulaci veřejných obrázků ve složkách uživatelského rozhraní. Tyto obrázky je možné kdykoliv otevírat a dále s nimi manipulovat, ale po určité době jsou rovněž automaticky smazány. Poslední adresář *private* obsahuje podadresář pro každého registrovaného uživatele. Zde se ukládají jeho soukromé obrázky, které nejsou promazávány. S limity pro vkládání fotek a podobné aktivity není uvažováno a zmíněný automatický skript pracuje až při nedostatku místa. Přesouvání mezi jednotlivými místy i pozdější uložení na disk je dovoleno, avšak další možnosti jsou předmětem budoucího rozšíření (viz 7).

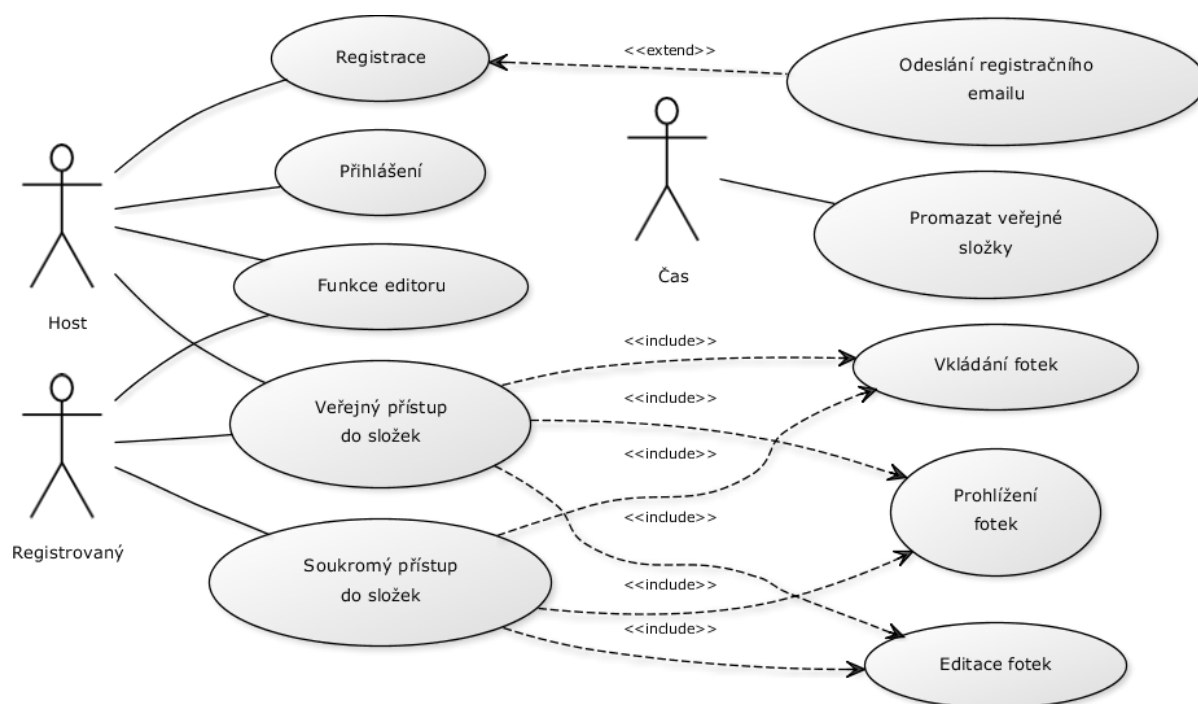
6.1.4 Editor

Návrh editoru vycházel ze zaběhnutého vzoru pracovní plochy, nabídky funkcí a panelu nástrojů rozšířený o přístup do složek upravených/neupravených fotek. V konečné fázi zůstal i informační panel, který byl původně vložen pouze dočasně pro zobrazování kontrolních informací. Nyní vypisuje průběh načítání a informace o vloženém obrázku. Panel nástrojů lze libovolně přesunovat a pomocí záchytných bodů (zobrazených při přesunování) také otáčet.

6.2 Model případů užití

Model případů užití je metoda softwarového inženýrství pro specifikaci a analýzu požadavků na aplikaci aniž by byla odhalena její vnitřní struktura. Graficky znázorňuje účastníky, případy užití, vzájemné interakce a hranice systému. [9] Byl vytvořen na základě návrhu vzhledu uživatelského rozhraní. Aktéry modelu jsou:

- *Host* – nepřihlášený návštěvník, který může se přihlásit, případně vytvořit účet. Má k dispozici všechny funkce editoru, má povolen přístup ke všem veřejným fotkám ve složkách, které může prohlížet, upravovat a následně opět uložit. Rovněž může veřejně vkládat nové fotky.
- *Uživatel* – disponuje všemi vlastnostmi *hosta* a navíc může složky využívat i soukromě tak, že k vloženým fotkám má přístup pouze on.
- *Čas* – automatická událost promazání veřejných fotek pro zajištění stálé dostupnosti služby. Není přesně definovaný a závisí na aktuálním stavu systému.

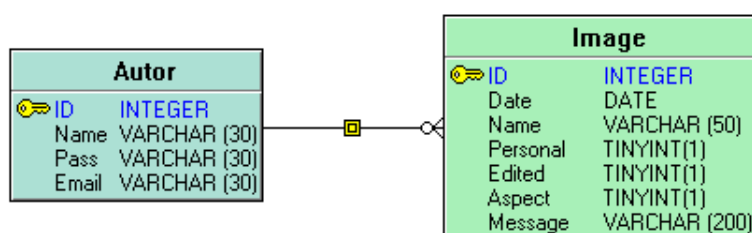


Obrázek 6.2: Diagram případů užití

6.3 Entity Relationship Diagram

Entity Relationship Diagram (ERD) slouží k modelování dat aplikační domény a jejich vztahů. Základními pojmy tohoto modelu jsou *entita* („věc“ reálného světa) a *vztah* (vyjadřuje asociaci mezi entitami). *Uzly* diagramu představují entitní množiny (množiny entit téhož typu) a *hrany* vztahy mezi nimi. Každá entitní množina obsahuje *atributy* (vlastnost entity, která nás zajímá) včetně kandidátních klíčů, z nichž jeden musí být primární. [9]

Diagram pro tento projekt je velice jednoduchý, obsahuje pouhé dvě entity a jeden vztah (viz Obrázek 6.3). Primárním klíčem je v obou případech automaticky inkrementovaný identifikátor ID. Cizí klíč zde není zobrazen. Atributy jsou použity v anglickém jazyce pro vyhnutí se implementačním komplikacím.



Obrázek 6.3: Jednoduchý Entity-Relationship Diagram

Autor

Entita zastupující registrovaného uživatele, který se může přihlásit a vkládat obrázky pod svým účtem. Pro přihlášení vyžaduje pouze jméno a heslo. Evidované údaje:

- *ID* – identifikační číslo uživatele, primární klíč
- *Name* – přihlašovací jméno
- *Pass* – přihlašovací heslo, šifrování se neuvažuje
- *Email* – kontaktní email, pro možný kontakt, ověřuje se syntaktická správnost

Image

Entita obsahující informace o daném obrázku. Samotné obrázky jsou ukládány externě (viz 6.1.3). Každý obrázek má svého autora. Pokud je vkládán anonymně, je přiřazen autorovi s identifikačním číslem 1 – *Anonym*. Evidované údaje:

- *ID* – identifikační číslo obrázku, primární klíč
- *Date* – datum modifikace pro řazení ve složkách
- *Name* – název obrázku
- *Personal* – rozlišení na veřejné/soukromé
- *Edited* – rozlišení na upravené/neupravené
- *Aspect* – poměr stran, rozlišení orientace obrázku pro zobrazení na úvodní stránce
- *Message* – připojená zpráva

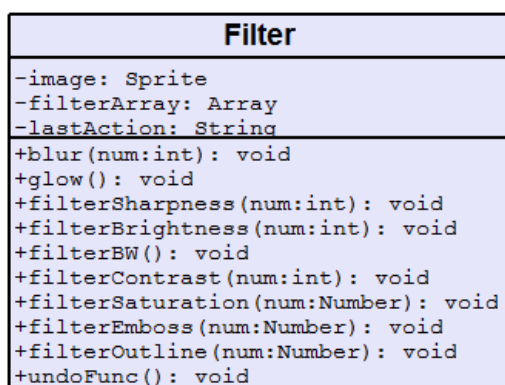
Umístění daného obrázku se nikde neuvádí a je složeno z výše zmíněných atributů. Pokud se jedná o veřejný obrázek, cesta je *images/public/\$Name*. V případě soukromého obrázku se zjistí jméno autora a cesta je *images/private/\$AutorName/\$Name*.

7 Implementace

Kapitola se zabývá popisem implementace celé aplikace, potažmo uživatelského rozhraní v jazyce ActionScript 3.0. Pro serverovou část bylo potřeba vytvořit několik PHP skriptů a MySQL dotazů nad databází. Pro spojení jednotlivých funkcí editoru často připadlo v úvahu více možných řešení, proto dále uvádím, které jsem vybral a proč.

7.1 Diagram tříd

Jazyk ActionScript patří od verze 2.0 do skupiny objektově orientovaných. Ačkoliv jsem se původně snažil psát zásadně objektově, tak se výsledná implementace často více podobá procedurální. To je zřejmě způsobeno menší praxí v oblasti objektového programování. Obrázek 7.1 obsahuje ukázkou třídy zajišťující aplikaci filtrů. Rozsáhlejší verze diagramu tříd je pak součástí přílohy (viz Příloha 1).



Obrázek 7.1: Třída pro operaci s filtry

Jednotlivé prvky editoru (pracovní plocha, tlačítka, okna pro vstupní hodnoty, atd.) jsou vytvořeny v grafickém prostředí Adobe Flash. To je spojeno s hlavní třídou *main*, která má přístup ke všem těmto prvkům a pomocí funkce *addEventListener* zpracovává jednotlivé události. Poskytuje tak mezivrstvu mezi editorem a samotnými metodami. Grafické operace jsou rozděleny do třech tříd na Filtry, Transformace a Kreslení. Pro implementaci jednotlivých metod je využito velmi rozšířených knihoven jazyka ActionScript. V těle metody se pak používá předdefinovaný typ pro danou operaci, nastavení jeho parametrů a aplikace na obrázek (viz Obrázek 7.2).

Provedení každé grafické operace se zaznamenává podle své třídy do speciálního pole kvůli realizaci funkce „krok zpět“. Jednotlivé třídy tuto možnost provádějí různě, proto je potřeba vědět, která metoda, respektive třída byla poslední volaná. Např. u třídy Filtrů jsou všechny aplikované filtry uloženy v jednom poli a pro krok zpět se na obrázek aplikuje pole filtrů bez toho naposledy vloženého.

Třídy *folders* a *login* se starají o zobrazení fotek ve složkách v závislosti na přihlášeném uživateli. Tyto jsou využity současně úvodní stránkou i editorem. Avšak pro zachování hezké URL jsem editor umístil zvlášť do adresáře *editor/*. A protože v třídě je potřeba uvést odkaz na původní soubor, musel jsem nechat obě výše zmíněné třídy duplicitní v obou adresářích.

```

public function filterSharpness(num:int):void
{
    var matrix:Array = [0, -1, 0,
                        -1, 5, -1,
                        0, -1, 0];

    var convolution:ConvolutionFilter = new ConvolutionFilter();
    convolution.matrixX = num;
    convolution.matrixY = num;
    convolution.matrix = matrix;
    convolution.divisor = 1;

    filterArray = image.filters;
    filterArray.push(convolution);
    image.filters = filterArray;
}

```

Obrázek 7.2: Metoda pro ostření obrazu

Jedním z požadavků na uživatelské rozhraní byla možnost při změně úrovně např. ostření, vidět výsledek ihned a nečekat až na potvrzení volby. Toho je dosaženo zavedením proměnné *lastAction*, která obsahuje poslední provedenou operaci a při opětovné změně parametrů se původní filtr odstraní a ihned aplikuje ten nový.

7.1.1 Externí třídy

Pro některé operace poskytuje jazyk ActionScript nedostatečné nebo žádné řešení, proto jsem využil externích tříd:

- *JPGEncoder*, *PNGEncoder* – třídy pro převod datového typu *BitmapData* do souborů JPEG a PNG. Přitom zajišťují rychlé a efektivní zpracování.
- *BitString* – jako součást dvou výše zmíněných.
- *Alert* – třída poskytující dialogové okna a zajišťující základní komunikaci s uživatelem. Umožňuje však pouze získání odpovědi v podobě ano/ne/zrušit. Pro zadávání hodnot od uživatele jsem vytvořil vlastní verzi, která z původní vychází.
- *ImageSaver* – slouží pro uložení obrázku nově vytvořeného v editoru na serveru.

7.2 Výchozí zobrazení

Popisuje způsob načtení obrázku do editoru a jeho další zpřístupnění. Zde totiž připadlo v úvahu více možných postupů. Prvním potencionálním řešením bylo vložit obrázek jako samostatný objekt jednoduše *addChild(img)*. S tímto přístupem vznikaly v průběhu implementace komplikace, proto se ukázalo jako lepší řešení vložit obrázek do jiného objektu *mainStage.addChild(img)* a pojmenovat si ho *img.name=jméno*. To se zužitkovalo při vytváření funkcí lupa a změna velikosti, pro které se využívá stejný postup. V případě lupy se transformace aplikuje na rodičovský objekt *mainStage.transform.matrix*. Obrázek se potom zmenší spolu s tímto objektem, ale jeho skutečná velikost zůstává stejná, zatímco při aplikaci přímo na potomka *mainStage.getChildByName(jméno).transform.matrix* se změní přímo jeho velikost.

V případě používání transformací a filtrů se tyto aplikují přímo na obrázek. Avšak u kreslení se opět nabízí více možností. Při kreslení do obrázku a následném použití např. některého filtru je tento

aplikován i na kreslenou část, ovšem další kreslení je již ovlivněno zmíněným filtrem. Druhá možnost kreslit do samostatné vrstvy problém neřeší. Optimální by byl způsob kombinace spojení s obrázkem a přidávání nových vrstev. Pro upřesnění připomenu, že editor neumožňuje práci s vrstvami obrazu. Pokud se zde zmiňuji o vrstvách, myslím tím jednotlivé potomky objektu, které se překrývají.

7.3 Komunikace se serverem

Pro komunikaci flash aplikace se serverem jsem využil dvě metody. První je prosté odeslání dat PHP skriptu pomocí funkce `sendToURL(data)`. Druhou možností je stejný způsob odeslání, ale poté očekávání odpovědi, a to pomocí funkce `load(data)`. Předávání dat probíhá pomocí metody POST, která data posílá v těle HTTP požadavku. K této komunikaci dochází hned v několika případech. Tím prvním je vždy po načtení aplikace ověření, zda již není uživatel přihlášen. Dále potom odeslání registrace, přihlášení/odhlášení uživatele, prohlížení složek obrázků a jejich načítání/ukládání.

Vyhodnocení dat provádí až daný PHP skript. Bylo by možné a nejspíše i více efektivní, aby se odeslání dat, např. při zadání příliš krátkého hesla vůbec neprovádělo a bylo na tuto skutečnost upozorněno ihned. Zde jsem však měl tuto kontrolu dat k dispozici z dřívějších projektů, proto jsem ji využil. Součástí komunikace je tak i stavový kód, který upozorňuje na případné chyby. Jeho přítomnost by byla nutná v každém případě, a to pro indikaci chyby vzniklé při komunikaci s databází, případně při přesouvání souborů.

7.4 Problémy s řešením

Problémy se vyskytovaly neustále a často jejich odstraněním vznikaly nové. Zdrojem chyb bylo většinou nedostatečné seznámení se s jazykem ActionScript, případně nutná změna struktury programu prováděná za běhu. Původní návrh se totiž při některých implementačních částech ukazoval jako nereálný nebo nevhodný. Některé problémy se podařilo elegantně vyřešit, ale některé zůstaly:

- Velké obrázky jsou automaticky při načtení zmenšeny na velikost plochy editoru, avšak při následném použití lupy se toto zmenšení nezapočítá a provádí se vůči původní velikosti. To však není přímo na škodu a nijak neovlivňuje další úpravy.
- Při načítání náhledů fotek ve složkách se načítají zbytečně v celém formátu a následně se musí zmenšit na zlomek své velikosti, což není ideální. Řešením by bylo ukládat k fotkám zvlášť jejich náhledy, ale to je komplikace navíc.
- Jedním z menších problémů bylo zachytávání událostí na objektech a vyvolání příslušných funkcí, jak správně zacházet s daným objektem. Např. při umožnění přesouvání panelu nástrojů tak, aby se přesouval pouze panel a nešlo z něj přesunout ikony. To však bylo vyřešeno vhodným pojmenováním a použitím regulárních výrazů.
- Problém se vyskytl při ukládání fotek na serveru. Zatímco nahrání fotky s velkými rozměry z lokálního disku pomocí funkce `move_uploaded_file` proběhne v pořádku, následné uložení do složky a tedy i kopírování pomocí funkce `copy` hlásí chybu.

8 Výsledek

Při tvorbě uživatelských rozhraní je jednou z nejdůležitějších částí projektu jeho prezentace, testování a vyhodnocení. Této části bude proto věnován patřičný prostor. „*Aby bylo možno posoudit, zda je uživatelské rozhraní navrženo správně, je třeba analyzovat dialog člověka se strojem...*“⁶

Pro tuto analýzu lze využít několika metod. Tou nejjednodušší je *Empirická* (experimentální) metoda, která nepotřebuje žádné znalosti o způsobu zpracování komunikace na straně člověka. Pro správné použití metody je potřeba: [3]

- *Správná formulace otázky* – pro seriózní analýzu je třeba přesně formulovat otázku.
- *Identifikace závislostí* – je potřeba určit, co může experiment zkreslit. Tím může být rozdílná zkušenost uživatelů, opakované provádění nebo závislost na konkrétním počítači.
- *Výběr vhodných proměnných* – ty mohou být závislé (přímo určeny otázkou) a nezávislé (čas, pozornost člověka).

Formou dotazníku tedy oslovím co nejvíce osob, aby otestovaly a zhodnotily celou aplikaci. V tomto případě je to potřeba, protože po několika měsíční práci na tomto projektu nejsem schopen objektivně posoudit intuitivnost jako člověk, který uvidí aplikaci poprvé. První část dotazníku se zaměří na uživatelskou přívětivost a schopnost se orientovat např. podle přesně zadaných kroků. V druhé části se bude volným způsobem testovat funkčnost editoru. Uživatel si vyzkouší práci podle svých představ a podá zpětnou vazbu.

8.1 Zhodnocení uživatelského rozhraní

Úvodní stránka

Tato stránka obsahuje několik funkčních bloků, které jsou optimálně uspořádány a tvoří tak přehledný celek. Prvky úvodní stránky jsou popsány podle přílohy (viz. Příloha 2).

1. *Registrace, přihlášení* – zobrazuje se zde přihlášený uživatel a tlačítka pro registraci a přihlášení/odhlášení.
2. *Automatické zobrazování fotek* – náhodně vybrané veřejné fotky ze složek dodávají úvodní stránce průběžně novou tvář. U fotek je již při ukládání zjištěna jejich orientace pro správné zobrazení na výšku/na šířku.
3. *Přístup do složek* – prohlížení aktuálního obsahu složek, fotky nelze z úvodu přímo otevřít.
4. *Kreslení úvodního obrázku* – designový doplněk úvodní stránky, kdy návštěvník může nakreslit libovolný obrázek. Ten se po spuštění editoru uloží a bude se zobrazovat všem dalším návštěvníkům. Někdy vzniká problém využívání cache paměti prohlížečem, proto nový obrázek není k dispozici ihned. Dostupný je pouze jeden štětec a paleta barev.
5. *Spuštění editoru, popis aplikace* – vstupní bod do editoru, popis hlavních dispozic aplikace.
6. *Zadání projektu, Facebook plug-in* – zápatí stránky, název tématu, kontakt na řešitele a vedoucího. Dále propojení se sociální sítí Facebook, kde je vytvořena fanouškovská stránka.

Editor

Výsledná podoba editoru odpovídá jeho návrhu (viz 6.1.4).

⁶ Ray E. Eberts: User Interface Design

8.2 Testování

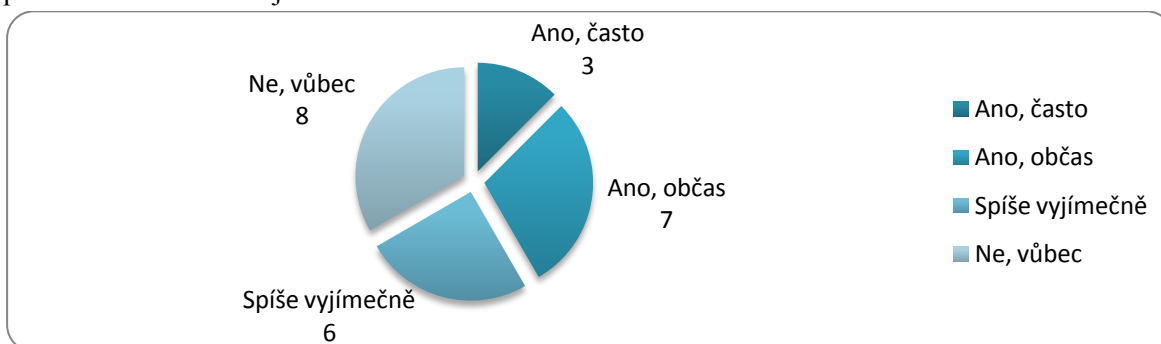
Při navrhování průběhu testování jsem vycházel z výše uvedených zásad. První otázkou bylo, komu bude test určen, aby byly kladené otázky a pokyny srozumitelné. Z konceptu aplikace vyplývá, že určitě není profesionálním editačním prostředím, jako spíše zábavnou a sociální službou. Proto budou otázky určené spíše pro méně zkušené až středně pokročilé uživatele. Aplikace nebyla během vývoje žádným způsobem publikována, tudíž nikdo nebyl zvýhodněn znalostí prostředí. Stejně tak je vyloučena závislost na konkrétním stroji, případně samotném prohlížeči, poněvadž Flash plug-in zaručuje vždy konzistentní zobrazení obsahu. Testování se zúčastnilo **24 lidí** různého věku a pohlaví.

8.2.1 Uživatelské testování

Tato část testu se skládá z obecných otázek na vztah uživatele k editaci obrazu, hodnocení uživatelského rozhraní a pochopení konceptu aplikace spolu s některými specifickými funkcemi. Každý bod začíná otázkou na uživatele a vysvětlením, proč je potřeba otázku položit. Následovat bude výsledný graf a vyhodnocení naměřených údajů.

1. Upravujete nějakým způsobem fotografie?

Úvodní obecná otázka zjišťuje zastoupení uživatelů, kteří pracují s obrazem a naopak, kteří se v této problematice neorientují.

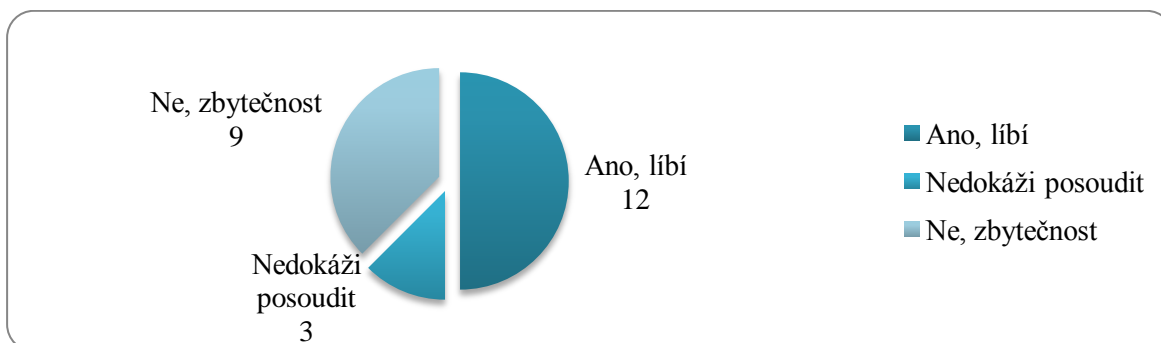


Obrázek 8.1: Otázka č. 1

Pokud budeme považovat tento reprezentativní vzorek za odpovídající vůči celé internetové komunitě, dovoluji si vyjádřit názor, že se jedná o pozitivní výsledek. Současně ukazuje na rovnoměrné zastoupení různých skupin uživatelů při testování.

2. Líbí se Vám možnost upravovat obrázky přímo na internetu?

Snahou je zjistit názor uživatelů na online aplikace.

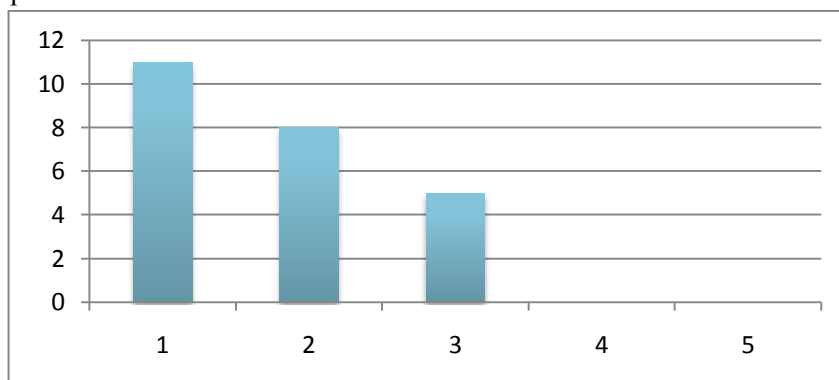


Obrázek 8.2: Otázka č. 2

Kladné hodnocení bylo často podmíněno možností spojení s jinou sítí, která již fotky obsahuje.

3. Oznámkuje (jako ve škole) vzhled aplikace.

První otázka zaměřená konkrétně na tento projekt. Jedná se o celkové hodnocení grafické podoby. Tento bod může být mírně zavádějící, pokud uživatel nemá přehled o možnostech a vzhledu jiných aplikací podobného charakteru.

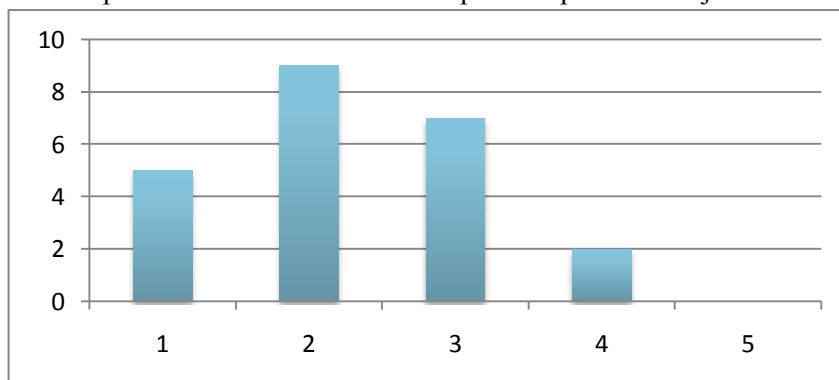


Obrázek 8.3: Otázka č. 3

Výsledek je vcelku pozitivní. Především úvodní stránka působí velmi příjemně, zatímco editor je hodnocen spíše průměrně. Zde se však dbalo hlavně na funkčnost.

4. Oznámkuje (jako ve škole) intuitivnost ovládání.

V tomto bodě se měl uživatel seznámit s prostředím, vyzkoušet možnosti aplikace a zhodnotit intuitivnost. Po samostatném prohlédnutí byly položeny doplňující otázky typu: Víte jak se registrovat? Jak používat obsah složek? Jak se používá panel nástrojů v editoru? Apod.

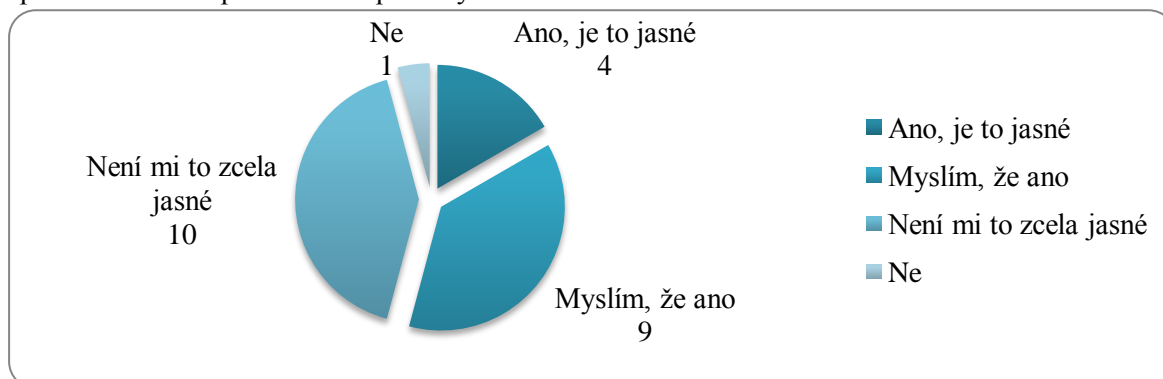


Obrázek 8.4: Otázka č. 4

Výsledkem je o poznání horší hodnocení, než u samotného vzhledu. Uživatel často dlouze bádá – co vlastně může udělat. Zde by bylo vhodné aplikaci doplnit o zobrazování doplňujících informací.

5. Chápete využití složek pro upravené/neupravené fotky?

Jedna z nejdůležitějších otázek. Toto propojení editoru se složkami je jednou z hlavních priorit aplikace. Uživatel se má orientovat v principu: Nová fotka patří do Neupravených fotek, následně se upraví v editoru a přesune do Upravených fotek.

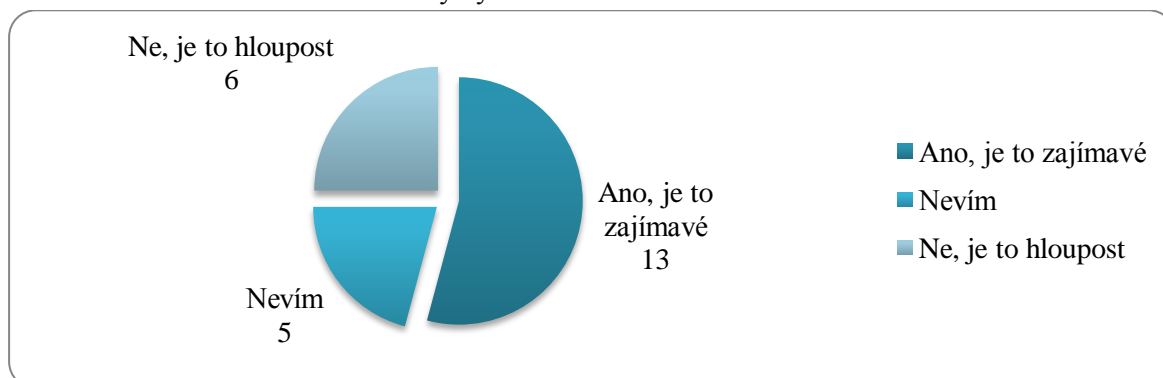


Obrázek 8.5: Otázka č. 5

Vnímání složek jako prostoru pro uložení fotografií bylo většinou jednoznačné, avšak uživatel nevěděl, jak ho používat. Proto by bylo rovněž vhodné u takto netradičních řešení uživateli přímo nabízet podrobnější popis.

6. Přejde Vám zajímavá možnost nechat upravit své fotky někým jiným a obdržet je pak automaticky na email?

Otázka má vystihovat celkový potenciál této myšlenky. Uživateli je princip funkce vysvětlen a názorně ukázán. Hodnotí tak skutečný význam.

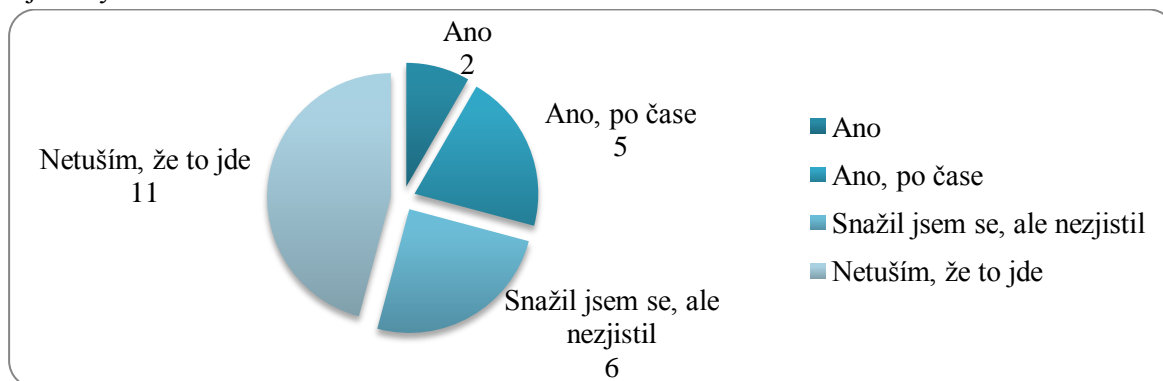


Obrázek 8.6: Otázka č. 6

Já osobně vnímám tuto myšlenku jako velmi zajímavou. Nejde přímo o zasílání na email, ale o celkový koncept předávání mezi více lidmi. Od uživatelů samozřejmě padla otázka „a proč by měl někdo chtít upravovat moje fotky?“. Na to jsem odpověděl jednoduše „a proč ne?“. Výsledek mě tak v celku potěšil.

7. Zjistil jste, jak nakreslit obrázek na úvodní stránce?

Tato možnost pro uživatele podílet se na vzhledu úvodní stránky není moc běžná a současně není nijak zvýrazněna.



Obrázek 8.7: Otázka č. 7

Očekávané negativní hodnocení se potvrdilo, zároveň po předvedení této možnosti byla hodnocena velice kladně. Opět je tedy potřeba dát o možnosti uživateli znát.

8.2.2 Funkční testování

V této části byl uživatel požádán o provedení přesného postupu při práci s editorem. Cílem bylo pozorovat orientaci v programu a celkovou délku provádění. Popsán bude postup jednotlivých kroků a poté souhrnně průběh a vyhodnocení testu.

• Test 1.

Otevření obrázku ze složky – zmenšení na 50% velikosti – přidání jasu.

První jednoduchý test, kdy uživatel zvolí fotku ze složky a jediným kliknutím ji otevře v editoru. Následně vybere obecně často používanou funkci pro změnu velikosti a na konec se musí zorientovat na panelu nástrojů pro vyhledání funkce jasu.

Průběh tohoto testu nepřinášel žádné komplikace, pouze v posledním bodě méně zkušený uživatelé váhali nad tlačítky panelu nástrojů a jejich identifikací.

• Test 2.

Otevření obrázku z počítače – rozostřit – zaostřit.

Zde si uživatel vyzkoušel použití vlastního obrázku.

Tento test nebyl pro nikoho problém a samotná práce uživatelů se výrazně zrychlila.

• Test 3.

Vytvořit nový obrázek – nakreslit libovolný obrazec – uložit obrázek do složky.

Zde místy vznikl problém ihned v počátku, kdy v editoru lze jen složitě rozeznat stav bez vytvořeného obrázku a s vytvořeným prázdným obrázkem. Po úspěšném vytvoření šlo již vše bez problémů.

Odhalené chyby

Během následného volného způsobu testování se podařilo uživatelům odhalit několik drobných chyb. Editor vykazoval při „správné“ kombinaci použitých funkcí funkcí nestandardní chování.

8.3 Porovnání

Protože všechny dříve zmíněné aplikace (viz 2) jsou vytvořeny pomocí platformy Flash, použiji nástroj WSMonitor⁷ pro zjištění využití paměti editorem během jeho používání.

	Picnik	Pixlr	Tento projekt
Spuštění editoru	70 MB	10 MB	10 MB
Nahrání obrázku	75 MB	15 MB	15 MB
Obrazové operace	110 MB	25 MB	20 MB

Tabulka 8.1: Sledování spotřeby paměti Flash aplikací

Na poli online editorů obrázků se však nepohybují pouze editory vytvořené platformou Flash. Vyberu tedy širší pole editorů a porovnáám jejich konkrétní vlastnosti. Počtem formátů se rozumí formát pro otevření/uložení. Výsledné bodové skóre počítám takto: 1 bod za každý formát, 1 bod za každou vlastnost. *v potaz jsou brány pouze volně dostupné funkce.

	Picnik	Pixlr	Phoenix	Thumba	MyImager	Tento projekt
Počet formátů	10/6	7/5	4/5	4/3	2/1	3/2
Technologie	Flash	Flash	HTML5	Silverlight	JavaScript	Flash
Jas, Kontrast	Ano	Ano	Ano	Ano	Ano	Ano
Odstín, Saturace	Ano	Ano	Ano	Ano	Ano	Ne
Vrstvy	Ne	Ano	Ano	Ne	Ne	Ne
Rotace, překlopení	Ano	Ano	Ano	Ano	Ano	Ano
Kouzelná hůlka	Ne	Ano	Ano	Ne	Ne	Ne
Klonovací razítko	Ne*	Ano	Ne	Ano	Ne	Ne
Úrovně	Ne*	Ano	Ano	Ne	Ano	Ne
Křivky	Ne*	Ano	Ne	Ne	Ne	Ne
Filtry	Ano	Ano	Ano	Ano	Ano	Ano
Kreslení	Ne	Ano	Ano	Ne	Ne	Ano
Sdílení	Ano	Ano	Ano	Ne	Ne	Ano
Účty	Ano	Ano	Ano	Ne	Ano	Ano
CELKEM	22 b.	25 b.	19 b.	12 b.	9 b.	11 b.

Tabulka 8.2: Konkrétní porovnání vlastností

⁷ dostupné na <http://www.websector.de/blog/2007/10/01/detecting-memory-leaks-in-flash-or-flex-applications-using-wsmonitor/>

9 Závěr

Práce je zaměřena na průběh vytváření online aplikace pro práci s obrazem. Popsáno bylo několik technologií a celkový teoretický základ. Bod zájmu však nestál pouze v samotné úpravě obrazu, ale v celkovém uživatelském rozhraní aplikace. Snažil jsem se vytvořit příjemné, intuitivní prostředí, které si návštěvník oblíbí. Účelem předkládané aplikace není ohromit svým technickým zpracováním, jako spíše nastínit alternativní možnosti pro sdílení a publikování, které jsou zde s jistou nadsázkou originální. Editor obsahuje především základní grafické funkce a je tak pro svou jednoduchost velice přehledný.

Pro mě osobně byla práce velmi přínosná, především blízké seznámení s platformou Flash vidím jako velkou výhodu do budoucna. Vyzkoušel jsem si spolupráci jazyka ActionScript a PHP, osvojil si práci s vektorovou grafikou a celkově si okusil průběh rozsáhlejšího projektu. Pro naučení jazyka jsem použil knihu *O'reilly Learning ActionScript 3.0 2nd Edition* psanou v anglickém jazyce, která se velmi dobře četla. Výhodou bylo také množství dostupných informací na internetu, mnoho ukázkových příkladů a řešení problémů na diskusních fórech.

Možností pro další vývoj je celá řada. Aplikace v současné podobě není schopna masového používání. Je potřeba rozšířit omezený prostor na serveru a zavést patřičné limity pro nahrávání fotek. Stejně tak je nutno doladit automatické promazávání podle provozu. Dále rozšířit správu uživatelských účtů a do složek zavést nabídku volitelného řazení a vyhledávání. Lepší možnost spojení se sociální sítí Facebook by pomohla většímu rozšíření. Pro úspěšnost aplikace však není hlavním cílem dát lidem výsledné upravené fotky, ale přitáhnout je k fotkám do editoru.

Literatura

- [1] Wikipedia: Webová aplikace [online], [rev. leden 2011] [cit. 2011-4-22].
URL: http://cs.wikipedia.org/wiki/Webová_aplikace
- [2] Burget R., Zeman D.: *Tvorba webových stránek (ITW)*. Brno: FIT VUT v Brně, 2006.
- [3] Zemčík P.: *Tvorba uživatelských rozhraní (ITU)*. Brno: FIT VUT v Brně, 2006,
Úvod, komunikace člověka se strojem.
- [4] Dvořák J.: *HTML 5: nová generace webů* [online], [rev. 2009-07-09], [cit. 2011-4-28].
URL: <http://www.zive.cz/clanky/html-5-nova-generace-webu/sc-3-a-147815/>
- [5] Zemčík P., Beran V.: *Zpracování obrazu (ZPO) přednášky*. Brno: FIT VUT v Brně, 2010, Úvod do zpracování obrazu, Lineární filtrace, Bodové transformace obrazu.
- [6] Hruška T., Burget R.: *Internetové aplikace (WAP) IV., část Programování serveru (PHP)*. Brno: FIT VUT v Brně, 2007.
- [7] Redakce Interval.cz: *Co je Silverlight?* [online], [rev. 2008-11-03], [cit. 2011-4-29].
URL: <http://www.interval.cz/clanky/silverlight-co-je-silverlight/>
- [8] Bruchanov M.: *Internetové Diskrétní 2D konvoluce*. [online] Praha: ČVUT FEL, 2006.
URL: <http://bruxy.regnet.cz/fel/36ACS/konvoluce.pdf>
- [9] Křena B., Kočí R.: *Úvod do softwarového inženýrství (IUS)*. Brno: FIT VUT v Brně, 2010.

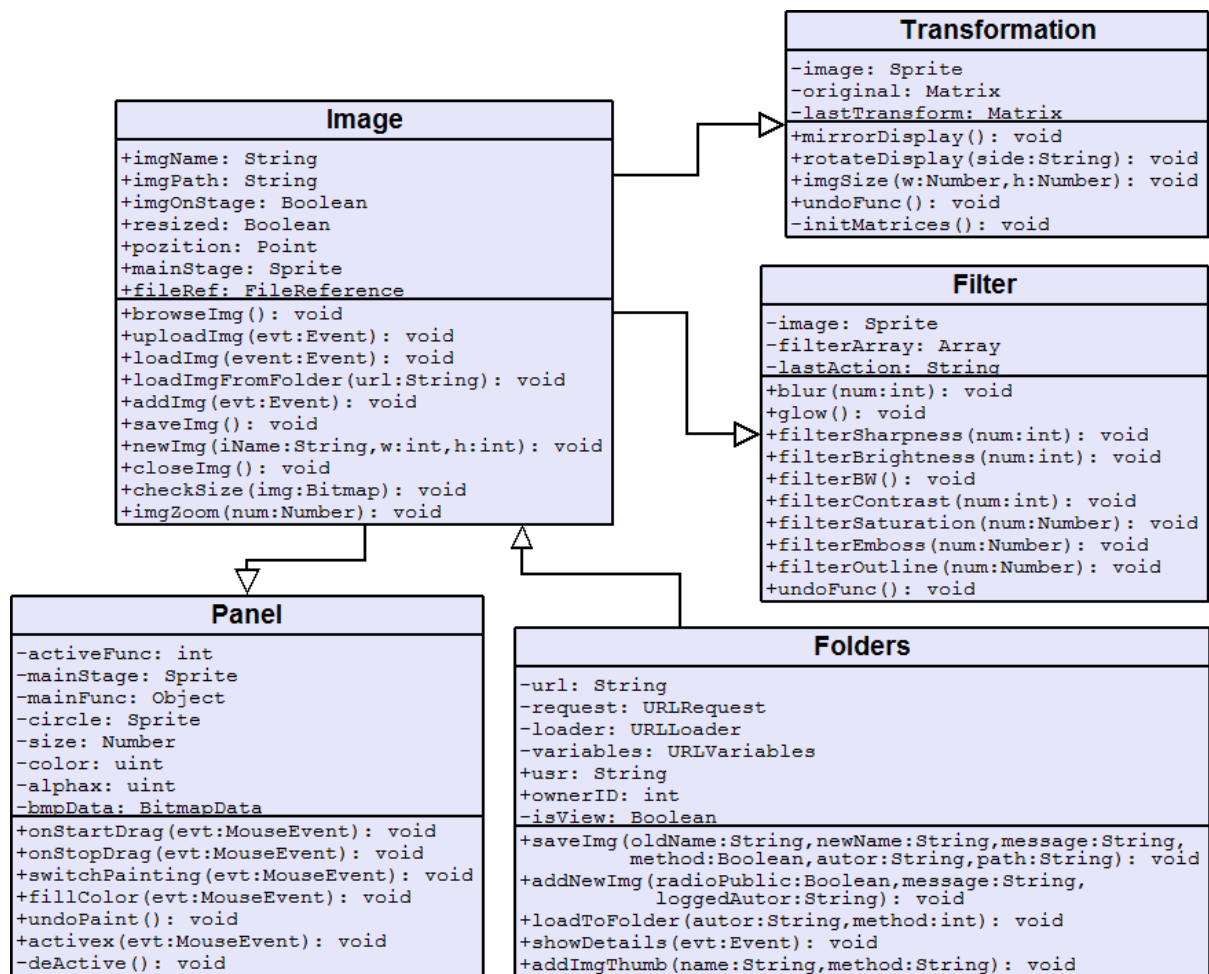
Seznam příloh

Příloha 1. Zjednodušený diagram tříd

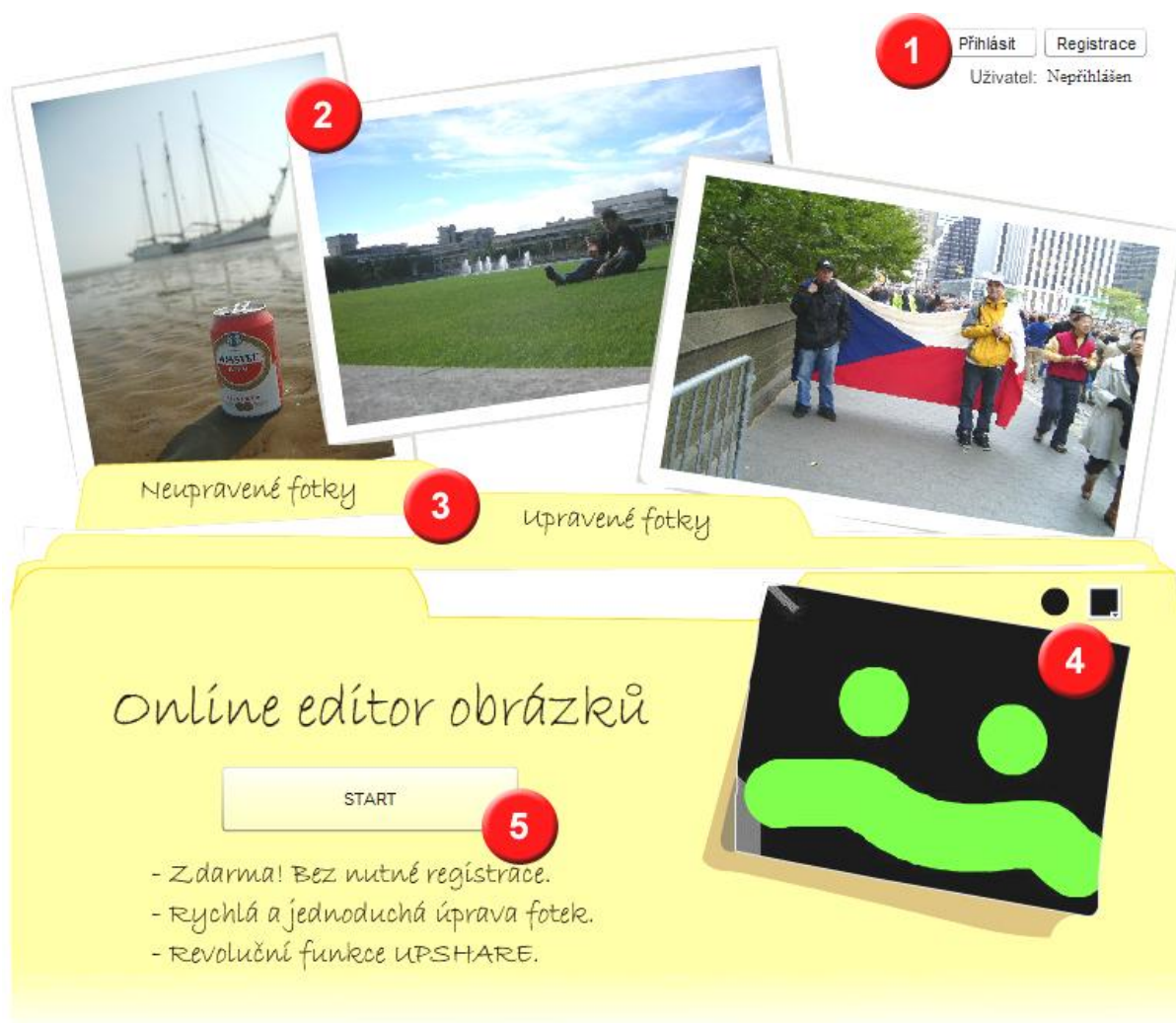
Příloha 2. Ukázka úvodní stránky

Příloha 3. CD se zdrojovými kódy aplikace a s elektronickou verzí této práce

Příloha 1



Příloha 2



Bakalářská práce na téma: **Online editor obrázků** | student: [Jakub Vodák](#) | vedoucí: [Ing. Pavel Svoboda](#) | © 2011

Like One person likes this.